SensCalc: user guide

Maksym Ovchynnikov

E-mail: o.maxim@gmail.com

Contents

Contents

1 General overview

Long-Lived Particles (LLPs) are a class of extensions of the Standard Model (SM) that add a particle with the lifetime τ_{LLP} that well exceed the lifetimes of unstable SM species, such as a pion or a muon. It means that once produced in collisions of the particles in laboratory, LLPs may travel macroscopic distances $c\tau_{\text{LLP}}\gamma v$ before decaying.
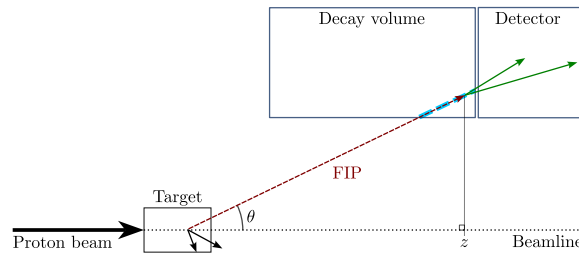

Figure 1. Setup of the experiment that may search for feebly-interacting particles (FIPs).

A typical experiment that may efficiently search for LLPs has the following setup (see Fig. 1):

1. A proton beam colliding with another proton beam or hitting a fixed target, where LLPs may be produced.

2. A decay/scattering volume displaced from the collision point, where the LLP would decay or scatter.

3. Depending on the setup, the detector system located downward the decay volume, to detect decay products.

Consider the signature when we search for a decay of a single LLP. Then, the number of events with decays of LLPs may be represented by the integral

$$N_{\text{events}} = \sum_i N^i_{\text{LLP,tot}} \times \int d\theta dE dz f^i_{\text{LLP}}(\theta, E) \epsilon_{\text{az}}(\theta, z) \frac{dP_{\text{decay}}}{dz} \cdot \text{Br}_{\text{vis}} \cdot \epsilon_{\text{dec}}(E, \theta, z) \qquad (1.1)$$

Here: i denotes the LLP production mode; $N^i_{\text{LLP,tot}}$ is the total number of LLPs produced in collisions; E, θ, z are the energy, polar angle, and longitudinal displacement of the decaying LLP (assuming that the z axis is aligned along the incoming beam); $f^i_{\text{LLP}}(\theta, E)$ is the angle-energy distribution of the LLP for this channel; $\epsilon_{\text{az}}(\theta, z)$ is the fraction of azimuthal angles ϕ for which the point $\{\phi, \theta, z\}$ at the fixed θ, z is inside the decay volume;

$$\frac{dP_{\text{decay}}}{dz} = \frac{\exp\left[-\frac{z}{\cos(\theta)c\tau_{\text{LLP}}\sqrt{E^2 - m^2}/m}\right]}{\cos(\theta)c\tau_{\text{LLP}}\sqrt{E^2 - m^2}/m} \qquad (1.2)$$

is the differential decay probability; Br_{vis} is the branching ratios of the visible decay modes (selected at the stage of notebook 1); $\epsilon_{\text{az}}(\theta, z)$ is the decay products acceptance – the fraction of events with LLPs decaying inside the decay volume, for which the events pass the selection (some minimal amount of decay products intersect the detector and obey various kinematic properties).

The code `SensCalc` consists of a few `Mathematica` notebooks that compute Eq. (1.1) for various LLPs (see Fig. 2):[1]

– 1. `Acceptances.nb` (computing $(\epsilon_{\text{az}}, \epsilon_{\text{dec}})$).

– 2. `LLP distribution.nb` (computing f^i_{LLP}).

– 3. `LLP sensitivity.nb`, computing the integral (1.1) for the given LLP `LLP` to get the tabulated number of events and then calculate the sensitivity

– 4. `Plots.nb`, computing the plots with sensitivities of various experiments.

On top of that, the recent updates have added the following notebooks:

– `EventCalc.nb` [arXiv:2409.11096], which generates the decay events of various LLPs in a format similar to the traditional Monte Carlo generators. It uses the input produced by the second notebook of `SensCalc`, `LLP distribution.nb`, as well as the routines utilized by the other notebooks. A python-based version of `EventCalc` is being used nowadays in the full simulation framework of the SHiP experiment.

– `Di-decays.nb` [arXiv:2503.01760], which generates the events where two LLPs are produced per event and then may both decay inside a decay volume.

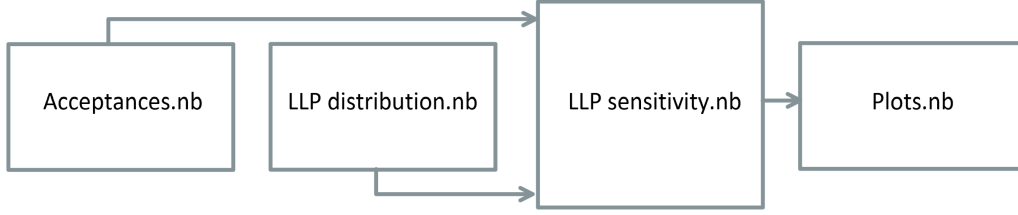[1]Contact the author: o.maxim@gmail.com

Figure 2. Description of the modular structure of `SensCalc`. The notebook `Acceptances.nb` produces the list of acceptances ϵ_{az} and ϵ_{dec} entering Eq. (2.1) of the accompanying paper for the selected experiment. The notebook `LLP distribution.nb` computes the distribution of LLPs $f(m, \theta, E)$ at the facility housing the experiment. The notebook `LLP sensitivity.nb` uses the input of the two previous notebooks to calculate the tabulated number of events, and calculates the sensitivity in the mass-coupling plane as a function of the remaining parameters, such as the minimal number of events and any additional model-specific parameters. Finally, `Plots.nb` produces the sensitivity plots from the output of the previous notebook.

1.1 Why `Mathematica`?

Evaluating sensitivities of lifetime frontier experiments to LLPs is often much simpler than detailed event analysis or background evaluation. If computing the distributions of mother particles producing the LLPs externally, the sensitivity evaluation is split into relatively trivial computations, such as calculating the LLP distribution function and evaluating the acceptances for the LLP and its decay products.

`Mathematica` is a great environment for such tasks. Namely, everything from computing the matrix elements of the LLP production and decay to working with the geometry of the experiment may be made inside it. This way, it is very suitable for people who are not familiar with special software used to compute sensitivities. In addition, `Mathematica` notebooks provide a convenient visual framework. Finally, there is a very friendly and experienced `Mathematica` community whose members may help with any question (and probably optimize the performance of `SensCalc` by another factor of ten).

Pure `Mathematica` does not suit well for performing some low-level operations such as producing the phase space of many-body decays or propagating the particles through space. Fortunately, to improve this part, the `Mathematica` code has been simply compiled into a native machine code, which may improve the performance by a factor of a hundred and more.

1.2 Installation

The user needs to have installed `Wolfram Mathematica`, version 14.1 or higher. In addition, the user must install the `Mathematica` package `FeynCalc` and a `C++` compiler recognized by `Mathematica`.[2] There are no other dependencies. The user can check whether the dependencies are properly installed by launching the notebook `0. Check for the needed soft.nb`.

1.3 Launching guidance

For the LLPs and experiments that are already implemented, the user just needs to launch the section *Launch this section to run the notebook* located at the top of each notebook. The relevant sections will then be launched automatically, and the notebook will prompt the user to specify the required inputs via dialog windows. If, however, users want to modify the model, geometry, or assumptions, they may change the code and inputs as described in the following sections.

---

[2]For Windows machines, a suitable choice is `Visual studio community`, where the package "Desktop development with C++" should be installed. For MacOS, the compiler should be recognized by default.

1.4 Performance

Performance significantly depends on CPU and LLP (multiplicity of its decay products, several choices of the LLP phenomenology description, the covered LLP mass range). On modern laptops, for most scenarios, the sensitivity may be computed from scratch (starting by computing the tabulated acceptances and ending by plotting the sensitivity) within less than an hour. Also, if being launched once for an experiment belonging to the given facility, the output of the notebook `2. LLP distribution.nb` may be used for any other experiment from the same facility, which reduces timing substantially. The main computational time is spent on calculating the decay products acceptance, which requires simulating the propagation of decay products per many LLPs decayed with different energies and at different points inside the decay volume. Performance may be improved in the future.

2 List of implemented LLPs

The currently implemented LLPs (and how they are called in `SensCalc` framework) are:

- The HNLs that are mixing with $\nu_e/\nu_\mu/\nu_\tau$. The names of these LLPs throughout the notebooks are `HNL-mixing-e`, `HNL-mixing-mu`, `HNL-mixing-tau`, or, collectively, `HNL`. The phenomenology of HNLs has been mostly taken from Ref. [1805.08567], except for smoothly matching between the hadronic decay widths of HNLs as described by perturbative QCD and exclusively (in terms of particular mesonic final state) separately for each mixing and neutral/charged currents.[3]

- Dark photons (named as `DP`). The phenomenology has been taken mostly from Refs. [2409.11096] and [2409.09123] (for describing the splitting functions and form-factors for the proton bremsstrahlung mode).

- Axion-Like Particles (ALPs):

    - Coupled solely to gluons at the scale $\Lambda = 1$ TeV (`ALP-gluon`). The model is also known as BC11, according to PBC definition [1901.09966].
    - Universally coupled to fermions at the scale $\Lambda = 1$ TeV (`ALP-fermion`). It is also known as BC10.
    - Coupled solely to photons (`ALP-photon`).

    More models of ALPs having hadronic couplings may be implemented with the help of the tool ALPs-phenomenology. The latter summarizes the phenomenology as described in [2501.04525] (see also the outdated study [2310.03524]). For the ALPs coupled to photons, the relevant phenomenology has been mostly taken from [1904.02091].

- Dark scalars with the quadratic hS and trilinear hSS couplings to the Higgs boson h (`Scalar`). The phenomenology has been mostly taken from [1904.10447] and [2407.13587] (theoretical uncertainties of the scalar decays into a pair of pions and kaons).

- Mediators coupled to the anomaly-free combinations of the baryon and lepton currents: $U_{\text{B-L}}(1), U_{B-3L_\mu}(1), U_{B-3L_e-L_\mu+L_\tau}(1)$, and $U_{B-L_e-3L_\mu+L_\tau}(1)$ (`B-L`, `B-3Lmu`, `B-3Le-Lmu+Ltau`, `B-Le-3Lmu+Ltau`). The phenomenology has been mostly taken from [1801.04847], with the uncertainty for the proton bremsstrahlung computed using the approach of Ref. [2409.11096].

---

[3]In the reference, the widths have been matched at $m_N \simeq 1$ GeV independently of the current type and mixing, whereas this is not correct for, e.g., the τ mixing, where the matching of the charged-current widths should be above $m_N = m_\tau$.

– Millicharged particles (MCP) (in public – only calculation of the production flux) (`MCP`). The phenomenology of MCPs has been mostly taken from Ref. [2205.09137].

– Inelastic dark matter coupled via the dipole electromagnetic operator with various mass splittings (`Inelastic-dipole-DM`) [2301.05252]. Other mass splitting cases may be included by launching the notebook `idm-pheno.nb` in the folder *phenomenology/Inelastic-dipole-DM*.

Definitions of the models and references on the implemented phenomenology are also written in the file `ModelDescriptors.mx`, located in the folder *phenomenology* of the repository.

In the future, I plan to add many other models, such as inelastic/elastic dark matter coupled via dark photons and other vector mediators, dipole portal of HNLs, and dark QCD models.

3 Notebook 1. Acceptances.nb

This notebook computes the tabulated acceptances $\epsilon_{\mathrm{az}}, \epsilon_{\mathrm{dec}}$ from Eq. (1.1). To do this, one needs to know the phenomenology of LLP decays: which decay modes it has, depending on the mass, the decay matrix elements governing the phase space of decay products. In general, we do not know the phenomenology perfectly, which results in theoretical uncertainties in the decay description (see, e.g., [2407.13587], [2503.01760]).

Consider an LLP with several descriptions of the decay phenomenology specified by {descr}.[4] The output of the first notebook is the following data, for each `descr`:

{ Facility, ECAL availability, acceptances data, table with experiment-related numbers,

branching ratio of visible LLP decay modes, explanation of descr, proper lifetime $c\tau_{\mathrm{LLP}}$}

The data is located in the folder
Acceptances/⟨Given experiment⟩
The description of the output:

1. The 1st, 2nd, 4th columns are experiment- or facility-specific details: at which facility the experiment is located (`Facility`), what is the number of proton-proton or proton-target collisions, and what are the production fractions of various SM particles such as mesons or heavy SM bosons (`table with experiment-related numbers`), whether the detector has ECAL (`ECAL availability`).

2. `acceptances data` is the table with the following columns:

$$\{m_{\mathrm{LLP}}, \theta, E, z, \epsilon_{\mathrm{az}}, \epsilon_{\mathrm{dec}}\} \qquad (3.1)$$

where m_{LLP} is the LLP's mass and E is its energy.

3. For the selected descriptions of the LLP decay phenomenology `descr`, the interpolations of the branching ratio of the LLP's decay channels users selected to calculate the decay products acceptance, $\mathrm{Br}_{\mathrm{vis}}(m_{\mathrm{LLP}})$ from Eq. (1.1) (`branching ratio of visible LLP decay modes`), and the LLP lifetime $c\tau_{\mathrm{LLP}}(m_{\mathrm{LLP}}, \mathrm{coupling})$.

3.1 How the notebook works

Below is a description of how the notebook works, assuming that all the required ingredients are provided (the requirements are given in the subsection below):

---

[4]The existence of several descriptions reflects the theoretical uncertainty in LLP's phenomenology.

1. First, users specify the experiment for which experiments they want to compute the acceptances (section `Choosing the experiment`), the list of the particles detectable at the experiment, and selection criteria for the decay products (section *Specifying cuts on the decay products*).

2. Next, the notebook computes the tabulated azimuthal acceptance $\epsilon_{\mathrm{az}}(\theta, z)$ (cf. *Azimuthal acceptance calculation*). It starts by evaluating the polar angle coverage of the decay volume and the detector, which is done automatically by discretizing the implemented geometry of the experiment. To ensure the correctness of the implementation of the experiment setup, the user should check the values of the parameters

   `ThetaDecVolGivenExperimentMin`,

   `ThetaDecVolGivenExperimentMax`,

   `ThetaDetGivenExperimentMin`,

   `ThetaDetGivenExperimentMax`,

   and the visualization of the decay volume and detector in section *Visualization of the geometry*. Afterward, the notebook computes the grid of polar angles θ and z within the decay volume's coverage. Next, for each set of (θ, z), the list of values of the azimuthal angle ϕ that belongs to the decay volume acceptance is generated.[5] Once the computation of the azimuthal acceptance has finished, the user should check that the volume of the decay volume matches the value obtained using the computed acceptance ϵ_{az} and Eq. (5) from the accompanying paper (cf. section *Grid with azimuthal acceptance*). Another cross-check is the visualization of the generated grid of the LLP's coordinates inside the decay volume, separately for the LLPs that point or do not point to the detector, and checking that the volume integral of the azimuthal acceptance matches with the total volume of the decay volume. These tests, together with displaying the E, θ, and z grids, are made in section *Azimuthal acceptance calculations*.

3. To compute the decay products acceptance, one must first simulate LLP decays at the rest frame of the decaying LLP. Consider a particle `particle` (which may be either an LLP or an SM particle) with mass `mparticle`. The phase space for the particular decay channel `process` is generated on-flight by the block

   `PhaseSpaceDecaysRest[particle,mparticle,process,Nevents]`

   defined in the notebook *llp-decays.nb* located in the folder *codes* and evaluated externally from the main notebook. Here, `Nevents` is the number of decay events to generate. If some of the LLP's decay products are unstable, it repeats the routine

   `PhaseSpaceStableFromUnstableBlock`

   which decays it until only (meta)stable products are left: $\gamma, \pi^{\pm}, K^{\pm}, K_L, e^{\pm}, \mu^{\pm}, \nu, p, n$. The phase space is organized in a table, with each row representing a particular decay. Every 8 elements of the given row represent a particular decay product, with the following meanings:

   `p_x p_y p_z Energy mass PDG-identifier electric-charge stability`

   where all dimensional units are in GeV, `pdg-identifier` is the PDG ID in the form `ID.` (e.g., `22.`, note the presence of the dot ".!"), electric charge is in units of the proton's charge, and `stability` for the (meta)-stable particles must be `1.` (and `0.` if the particle is unstable).

---

[5]Suppose the decay volume is not itself the detector. In that case, the notebook computes two grids of ϕ: one for when the LLP decays inside the decay volume and simultaneously points to the end of the detector (`EpsilonAzPhiListToDet`), and another one for when the LLP decays inside but does not point to it (`EpsilonAzPhiListNotToDet`).

The block `PhaseSpaceStableFromUnstableBlock` treats decays into partons (such as LLP $\to$ $u\bar{u}, c\bar{c}$, etc.) as decays into a pair of the lightest charged mesons containing the given parton.

Alternatively, if the given decay is into partons, instead of generating on-flight, it may be possible to use the phase space pre-generated in `MadGraph5+pythia8` for several characteristic masses `mLLP`. In this case, the partons are replaced with a bunch of hadrons appearing because of the showering and hadronization. The hadronized phase space computed at the rest frame of the decaying LLP is called by the block

`PhaseSpaceDecaysRestJets[LLP, process, mLLP]`

defined in the notebook `llp-decays-jets.nb`. The UFO files used to generate the phase space are located in the folder *UFO files* (the model `fips-jets`). The phase space is stored in the folder

simulated phase space

The names of the files must have the pattern

`<LLP>_<process>_mass.m`

where `<LLP>`, `<process>` must exactly match the names of the LLP and the given decay process into the partons as defined in the notebook. An example of the filename is

`Scalar_Jets-cc_3.8.m`,

which contains the phase space of the decay of the Higgs-like scalar with mass 3.8 GeV into hadrons resulting from the jets $c\bar{c}$. The structure of the phase space must be the same as the one generated by `PhaseSpaceDecaysRest`. To have a rectangular array where all the rows have the same length, every row is padded from the right with `-999.` if needed.

4. When computing the decay products acceptance (cf. *Decay acceptance calculation*), the notebook simulates the phase space of the relevant decay channels and calculates the decay acceptance for several signatures. All the definitions are loaded from an external notebook `decay-acceptance.nb`. In particular, for the given LLP mass, the tabulated decay acceptance is calculated by the block

`FinalBlockMass[LLP, mLLP, proclist, isim, HadronizationOption,`

`acceptancetypechoice, ntracksmin, decaydescr]`

Here, `proclist` is the list of the processes used to compute the decay products acceptance (the users are asked about the list of these processes right before the computation); `isim` is the number of decays simulated per the ALP mass `mLLP`, longitudinal displacement from the production point `zLLP`, the polar angle `thetaLLP`, the azimuthal angle `phiLLP`, and the energy `ELLP`; the optimal value of `isim` is around 1000. `HadronizationOption` may be `True` of `False` depending on whether the user wants to use the pre-simulated hadronized decay phase space of LLPs treat them in a simplified manner;[6]. `acceptancetypechoice` describes the signature used to calculate the decay products acceptance:

   – `>=2`, when at least two decay products with total zero electric charge must point to the detector and satisfy cuts on kinematics.

   – `All detectable`, when all detectable particles must point to the detector.

   – Others, specified in the notebook. Their definition may be seen using the commands
     `a2 = Keys[DownValues@explanationDecayAcceptance][[All, 1, 1]];`
     `{#,explanationDecayAcceptance[#]} &/@ a2`

---

[6]Importantly, for the decay into jets, if `HadronizationOption` is `True`, one may choose only the masses `mLLP` for which `PhaseSpaceDecaysRestJets` contains the data. The notebook automatically computes such a mass list.

Finally, `decdescr` is the selected description of the LLP decays. The list of decay descriptions and their meaning for the given `LLP` may be accessed using

`{#, DecayDescriptionExplanation[LLP, #]} & /@ listDecayDescriptions[LLP]`

The block first evaluates the purely geometric part of the decay acceptance by requiring the projection of the trajectory on the final plane of the detector to be within the detector cross-section (the routine `conditionDecayAcceptanceGeometric`). If a dipole magnet is placed somewhere, then "kicks" to the trajectory, and the momentum components of the charged decay product in the direction transverse to the magnetic field are applied at the end of the magnet. Afterward, if the decay products are within the geometric acceptance, it calculates the acceptance for the other cuts, such as the energy cut, etc. (see the subsection *Blocks computing various cuts* from the external notebook), which serves as the event pre-selection. The experimental resolutions, such as Gaussian smearings of the energy and direction of motion, may be easily incorporated.

Within the routine, the decay products acceptance is first computed for the values of ϕ for which the LLP is inside the decay volume, and then averaged over ϕ, with each value being weighted by the corresponding azimuthal acceptance. Currently, `SensCalc` may generate on-flight 2-body, 3-body, and 4-body decays of LLPs (of course, if they contain unstable particles, they are decayed recursively). However, generic n-body decays may be implemented completely similarly to the 4-body decays.

Depending on the specific LLP, experiment, OS, and CPU, the notebook runs for $\mathcal{O}(1\text{ hour})$ (typically much faster) per LLP. The runtime also significantly depends on the grid density and the number of simulated decays; the latter is parameterized by the number of ϕ values considered when evaluating the decay products acceptance (`NofPhiVals`), and by the number of decays simulated for each (m, E, θ, z) point (`ivaltest`).

3.2 What users have to provide

If the user wants to implement a new LLP, they need to provide the following:

- Set of decay descriptions of the LLP {`decdescr`} and, for each of them, the description. `DecayDescriptionExplanation[LLP, decdescr]`.

- For each decay description `decdescr`, the lifetime `cτLLP[LLP, decdescr, mLLP, coupling]`, where `coupling` is the traditional label of the coupling for the given LLP (say, for dark photons, it is $\epsilon 2$), that has to be also defined via `couplingSymbol[LLPsel]`.

- For each decay description `decdescr`, the interpolated branching ratios of the decay processes

  `BrRatiosList[LLP, mLLP, process,decdescr]`

  where `process` is the given decay process, `mLLP` is the LLP mass in GeV, and the associated list of the decay products decay channels `ListDecayProducts[LLP, process]` in the form {`product1`, `product2`, `product3`, `product4`}. If the processes are 2- or 3-body, the remaining products must be replaced with `"Null"`. The list of recognizable products is given in section *Properties of SM particles*. For instance, for a 3-body decay into $\pi^+\pi^-\gamma$, the list of decay products is {π^+,π^-,γ,`"Null"`}.[7]

- The squared matrix elements of the decay at the rest frame of the decaying particle. They are needed only for n-body decays with $n \geq 3$. Currently, `SensCalc` may only include it for

---

[7]If the user instead wants to use the phase space pre-generated in `PYTHIA8`, the list of decay products is not needed.

3-body decays; for decays with higher multiplicities, a unit matrix element is assumed.[8] For 3-body decays, the squared matrix elements must depend on the decaying LLP mass `m`, the energy of the 1st particle `E1`, the energy of the 3rd particle `E3`. The user may enter the matrix element and calculate its square using `FeynCalc`. The list of all squared matrix elements is given by

`Msquared3BodyLLP[LLP, process, E1, E3, mLLP]`

– For decays into partons (e.g., into quark-antiquark pair), the pre-simulated phase space of decay to be called by `PhaseSpaceDecaysRestJets[LLP, process, mLLP]`.

– The list of LLP masses for which the tabulated acceptance will be computed. It is called by `mLLPlist[LLP]`; for the implemented LLPs, it is defined in section `Mass range for the acceptance computation` of the main notebook. Above the masses where decays into partons become possible and if the user wants to use the phase space pre-generated in `PYTHIA8`, the mass list must only contain the masses for which `PhaseSpaceDecaysRestJets[LLP, process, mLLP]` is available. Otherwise, the routine will switch to computing the acceptance using the non-hadronized phase space.

For the implemented models, the branching ratios, lifetimes, and squared matrix elements are stored in the folder
phenomenology/LLP/decay widths
All these definitions are loaded by calling the notebooks `<LLP>.nb` located in the folder *codes/Acceptances.*
For the computation of the azimuthal and decay acceptances, the user has to provide:

1. The geometry and dimensions of the decay volume and detector, as well as the properties of the detector. For the detector equipment, the information that currently needs to be provided is the presence of the ECAL and of the dipole magnet, their positions, and the mean magnetic field of the magnet. This is done in the section *Geometry of different experiments and relevant cross-sections*, which loads the external notebook `experiments.nb`.

   – If the decay volume and detector have a simple shape – box, cylinder, or annular cylinder, the only required inputs are the parameters describing the dimensions; they are listed in the subsection *Geometry* belonging to the subsection of the relevant experiment. An example of such parameters is `zToDecayVolumeExperiment[<Experiment>]`
   which specifies the longitudinal distance from the collision point to the beginning of the decay volume. If the detector has a more complicated shape, the user should implement its geometry in the section *Full geometry of the decay volume and detector*. Examples of already implemented non-standard geometries include the SHiP experiment and LHCb.[9] Regardless of how they were implemented, all the geometries are finally added to unified lists. For instance,
   `DecayVolumeGeometry[experiment]`
   contains the full geometry of the decay volume.

2. The integrated luminosity (for colliders) or the number of protons on target (for beam dump experiments) and the relevant production cross-section for the Standard Model particles (cf. subsection *Cross-sections* of the external notebook `experiments.nb`).

---

[8]To be implemented in the future.

[9]Another example is the geometry of the ANUBIS experiment in the ceiling configuration; it is more complicated and hence stored in a different notebook `1. Acceptances. ANUBIS-ceiling`.

3. Conditions for the LLP to decay inside the decay volume and for the decay product to point/not point to the detector. These are

`IfLLPinsideDecVol[zLLP, xLLP, yLLP, experiment]`

`IfLLPtoDet[x1LLPproj, x2LLPproj, experiment]`

`IfLLPnotToDet[x1LLPproj, x2LLPproj, experiment]`

where `zLLP,xLLP,yLLP` are the coordinates of the decaying LLP (in meters), and `x1LLPproj`, `x2LLPproj` are the projections of particle's (LLP or its decay product) trajectory given its production/decay point and the presence/absence of the dipole magnet. `x1LLPproj` coincides with the x coordinate, while `x2LLPproj` may be the y coordinate (if the detector plane is transverse to the proton beam axis, as, e.g., for the SHiP experiment) or the z coordinate (in the other cases).

For the experiments with a simple shape of the decay volume and detector (e.g., box, cylinder, annular cylinder), these conditions may be calculated automatically by using the routines

`IfLLPinsideDecVolSimple[experiment]`

`IfParticlePointsToDetSimple[experiment]`

For more complicated cases, this has to be defined manually (see subsections *Conditions for the LLP to decay inside the decay volume and for the products to point to the end of the detector*).

4   Notebook 2. LLP distribution.nb

The output of the second notebook is a set of the tabulated LLP angle-energy distribution f_{LLP}^{i} from Eq. (1.1) of the form

$$\{m_{\mathrm{LLP}}, \theta, E, f_{\mathrm{LLP}}^{i}\}, \tag{4.1}$$

where the last column is the value of the distribution function for the given $(m_{\mathrm{LLP}}, \theta, E)$, and the production mechanism i. The distribution is normalized to one. The tables are located in the following folder:

   spectra/New physics particles spectra/⟨LLP⟩/⟨facility⟩/

In the case when the production is via scatterings, the output also includes the tabulated production probability, stored in the folder

   phenomenology/⟨LLP⟩/Production probabilities

Depending on the specific LLP, OS, CPU, and the number of generated decays, the notebook runs for $< \mathcal{O}(20)$ minutes.

Often, if the notebook has already been run once for a given facility, it does not need to be launched again for the other experiments housed at the same facility, as the shape of LLP's distribution is determined mainly by the beam energy, while the normalization may be applied multiplicatively.

When launching the notebook, the user is prompted to choose the LLP and the facility at which the considered experiment is located. The notebook then computes the distributions of the LLPs produced by all implemented channels or by the channels selected by the user by evaluating the chapter *Choose LLP and run*. The code that executes the computation of the tabulated distribution for the given LLP `LLP` and process `proc` must located in the section with the name *LLP_proc* (for example, the code computing the distribution of HNLs produced by decays of W bosons is located in section *HNL_W*).

For the LLPs produced in decays, the calculation is organized as follows. The notebook first samples the 4-momenta of mother particles according to their angle-energy distributions.[10] Then, it generates the tabulated angle-energy distribution of the LLP using the routine[11], which is systematically done with the help of the routine

   `BlockTabulatedPDFsFromDecaysMass[LLP, Facility, mLLP, process, nsim, thetalist, Elist, thetaminExtr, IfTesting]`

Here, `thetalist`, `Elist` are the lists of the polar angles and LLP energies suitable for the given production channel and facility used for the tabulation, `thetaminextr` is the angle below which the LLP's distribution f_{LLP} is assumed to have the scaling

$$\frac{d^2 f_{\text{LLP}}}{d\theta dE} \approx \frac{\sin(\theta)}{\sin(\theta_{\text{min,extr}})} \left(\frac{d^2 f_{\text{LLP}}}{d\theta dE} \right)_{\theta = \theta_{\text{min,extr}}} \tag{4.2}$$

to avoid the lack of statistics.

The other implemented production channels include quasi-elastic scatterings (examples are the Primakov process and photon fusion for the ALPs), proton bremsstrahlung, and mixing with neutral particles. They are handled by the following blocks:

- `BlockTabulatedPDFsFromPrimakov[Facility, Nucleus, thetalist, Elist]` for the production of ALPs in the Primakov process.

- `BlockTabulatedPDFsFromFusion[Facility, Nucleus, thetalist, Elist]` for the production of ALPs in the photon fusion, where `Nucleus` corresponds to the target nucleus.

- `BlockTabulatedPDFsFromBrem[LLP, Facility,Description]` for the proton bremsstrahlung, where `Description` is the description of the bremsstrahlung channel (explained in the accompanying paper about the revision of the dark photon parameter space). This block produces not only the tabulated distribution, but also computes the probability of producing LLP per bremsstrahlung, putting it to

  phenomenology/⟨LLP⟩/Production probabilities

  and does this for several choices of the parameters specifying the production, to account for theoretical uncertainties (see Ref. [2409.11096]).

- `BlockTabulatedPDFsFromMixingsMass[LLP, Facility, mLLP, process, thetalist, Elist, thetaminExtr]` for the (outdated but still widely used) production via mixing with various mesons.

Some of the processes cannot be properly generated inside `SensCalc`. Examples are deep inelastic scattering, important for the millicharged particles, dark photons and other vector mediators, and ALPs. Instead, the tabulated angle-energy distributions are pre-generated using `MadGraph`+`pythia8` and stored, for each of the LLPs, in the folder

   spectra/New physics particles spectra/⟨LLP⟩/Pregenerated

The UFO files used to generate the distributions of LLPs from the DIS processes (the model `fips-jets`) are located in the folder *UFO files*.

---

[10]See the routine `BlockPointsFromSmoothDistribution` from the external notebook `mother-sampler.nb` located in the folder *codes/LLP distribution*). For this, it imports the distribution of the mother particle `particle` at the given facility `facility` from Sec. *Preliminary code: mother particles sampling* and accordingly generates `nsim` random mother 4-momenta. This is done by the routine `PointsSampler[Facility,particle, nsim, IfTwo]`. The last argument, `"True"` or `"False"`, tells whether there are two LLPs per single decay of `particle`.

[11]The routines are located in the notebook `llp-sampler-decays.nb` located in the folder *codes/LLP distribution*. First, the notebook samples the phase space of the LLPs in the rest frame of the decaying particles (e.g., the routine `ThreeBodyDecaysEventsAtRestLLPprod[LLP, process, mLLP, nsim]` for 3-body decays), boosts them (`BoostedLLPsBlock[LLP, facility, mLLP, process, nsim]`), and finally produces a smooth LLP angle-energy distribution based on the generated data `FromDataToSmoothDistribution[boosteddata, mLLP, facility,IfTesting]`.

4.1 What users have to provide

The users have to provide:

1. The tabulated distributions of the mother particles in the form `polar angle in rad`, `Energy in GeV`, `Value of the distribution`. The distributions are located in the folder

   spectra/SM particles/

   They must be normalized by one. To be found and imported by the notebook, the filename of the given distribution must match the pattern

   `DoubleDistr_<facility>_<mother>.txt`

   The names of the mother particles recognizable by the notebook may be found in the output cell of the section

   `Masses and other parameters of SM particles`

   Alternatively, it may be accessed by `Keys[DownValues@mSM][[All, 1, 1]]`.

2. The information about the LLP production phenomenology. For the implemented LLPs, it is stored in the notebooks `prod-pheno-<LLP>.nb` located in the folder *codes/LLP distribution*. The minimal information includes: the name of the process (to be added to the list `ProcessesListAll[<LLP>]`); the production type for the given process

   `ProductionType[<LLP>,<process>]`

   which may be "Decay", "Mixing" (for the outdated description of the production via the mixing with neutral particles), or "Scattering" (if the production is via the scatterings); the list of particles participating in the production process

   `ReactionProductsList[<LLP>, <process>]`.[12]

   The description of each of the implemented production channels may be added as and later accessed using `DescriptionProcess[LLP, process]`.

   If the decay process is a 3-body, then one needs to add the squared matrix element

   `Msquared3Body[<LLP>, <proc>, mLLP, E1, E3]`

   of the process expressed in terms of the LLP mass and the energies `E1,E3` of the 1st and 3rd decay products, according to their ordering in `ReactionProductsList[<LLP>, <process>]`. If the same decaying particle can produce the LLP through multiple channels, one requires the branching ratios of these various production channels.[13]

3. If the given LLP production process is a 3-body decay, the matrix elements of, cf. section M^2 *of 3-body production for different LLPs*). The implemented The branching ratios for the implemented LLPs are located in the folder

   phenomenology/⟨LLP⟩/branching ratios/

   Users can add their branching ratios in any format, as long as they are imported appropriately.

---

[12]The first particle is always the mother product, and by convention, LLPs must be at the end of this list. For example, for the production process `B-3-body`, when HNLs are created by 3-body decays of B mesons, it is `{"Bplus", "Dstar", "e", "HNL-mixing-e"}`. Another example is proton bremsstrahlung, where the list looks as, e.g., `{"p","DP"}`.

[13]The implemented examples include HNLs (see the notebook `prod-pheno-HNL.nb` and the routine `BoostedHNLsBlock[LLP, Facility, mLLP, process, nsim]` in `llp-sampler-decays.nb`) and ALPs (`prod-pheno-ALP-fermion.nb` and `BoostedALPsBlock[LLP, Facility, mN, process, Nevents]` in `llp-sampler-decays.nb`).

5 Notebook 3. LLP sensitivity.nb

5.1 What the notebook does

Once the user specifies the experiment and LLP (section *Specifying the experiment*), the notebook interpolates the tabulated functions obtained by the previous notebook and constructs the integral of their product to obtain the number of events (1.1) as a function of the LLP's mass and coupling (section *Number of events*).

The main routine for computing the number of events for the LLP produced by the production channel `ProdChannel` is

`NeventsDiscret[`m_{LLP}`, ProdChannel, couplingslist]`

where `couplingslist` is the list of the values of the coupling in the format {`coupling1, coupling2, ...` }.

By default, the decay products acceptance is included in the calculation in the block `NeventsDiscret`. Alternatively, it is possible to set the decay products acceptance to 1 when computing the number of events by changing the line

`integrandtab = TableIntegrandDiscret[m, ProdChannel, "True"];`

to

`integrandtab = TableIntegrandDiscret[m, ProdChannel, "False"];`

in `NeventsDiscret[m, ProdChannel, couplingslist]`.

The output with the tabulated number of events (generated in the section *Exporting tabulated number of events*) for each of the production channels has the form

`'mass in GeV' 'coupling' 'N_Events(mass, coupling)'`

and is stored in the file with the name

`Nevents_⟨LLP⟩_⟨prod⟩_at_⟨experiment⟩_Npot=XXX_decayDescr=⟨descr⟩.dat`

located in the directory

Tabulated Nevents/⟨LLP⟩/⟨experiment⟩/

In the filename, `prod` means the production channel, XXX the number of proton (electron) collisions corresponding to the given experiment,[14] and `descr` is the description of LLP's decays.

Finally, the section *Computing sensitivities* imports the tabulated numbers of events, sums them over the production channels, and computes the sensitivity. The user must specify the minimum number of events and some LLP-specific parameters. The sensitivity curves are located in the folder

Sensitivity domains/⟨LLP⟩/⟨experiment⟩/

There are two files storing the sensitivities:

– `.mx`, in the form of a table

   `production channels, bremsstrahlung uncertainty, Drell-Yan uncertainty,`

   `decay description, table of sensitivity regions`

   where `production channels` is the list of the production channels selected to compute the sensitivity; `bremsstrahlung uncertainty, Drell-Yan uncertainty` are (optional) parameters defining the description of bremsstrahlung and Drell-Yan production channels within the uncertainty band; `decay description` is the description of the LLP's decay phenomenology; `table of sensitivity regions` is the table of 2D tables with disconnected sensitivity regions of the given experiment.

– `.json`, with the same content, but organized in the form of associations, e.g., `Production modes → production channels, ...`

---

[14]It may be easily rescaled at the stage of computing the sensitivities

The files have the name pattern

    `Sensitivity_<LLP>_at_<experiment>_Nev=XX_Npot=XX.mx`

where `Nev` is the minimal number of events required for the sensitivity, and `Npot` is the number of proton (electron) collisions. The files are located in the folder

    `Sensitivity domains/⟨LLP⟩/⟨experiment⟩`

Additionally, the same section computes the uncertainty in the sensitivity coming from the theoretical uncertainty in the production and decay modes. The uncertainty files have the name format

    `Uncertainty_<LLP>_at_<experiment>_Nev=XX_Npot=XX.mx`

and are located in the same folder as the sensitivity regions, while their content (for the .mx file) is

    `source of uncertainty, central sensitivity, uncertainty regions`

with `source of uncertainty` being "Production", "Decay", and "All" (the latter combines the uncertainty from the production and decay), `central sensitivity` the sensitivity assuming the central description of production and decays, while `uncertainty regions` the domains around the central sensitivity defining the uncertainty.

The typical running time to compute the notebook per experiment is $\mathcal{O}(5 - 20)$ minutes, depending on the variations of the bremsstrahlung and Drell-Yan process, and the number of production modes.

5.1.1 Generalized acceptances

In each notebook `3. LLP sensitivity.nb`, the section *Generalized acceptances* (or a similar one) produces the acceptances that give a qualitative understanding of the contribution of the factors entering the number of events (1.1) (the LLP distribution function, the azimuthal acceptance, the decay probability, the decay products acceptance). Namely, consider the lower bound of the sensitivity, where typically $c\tau_{\text{LLP}}\gamma \gg z_{\text{max}}$, with z_{max} being the maximal longitudinal coordinate of the decay volume. The number of events becomes

$$N_{\text{ev}} = \frac{N_{\text{prod}}}{c\tau_{\text{LLP}}} \times \int d\theta dE dz f_{\text{LLP}} \epsilon_{\text{az}} c\tau_{\text{LLP}} \frac{dP_{\text{decay}}}{dz} \times \epsilon_{\text{dec}} \tag{5.1}$$

The integral is the LLP lifetime-independent, as the differential decay probability can be approximated by $dP_{\text{decay}}/dz \approx 1/c\tau_{\text{LLP}}\sqrt{\gamma^2 - 1}\cos(\theta)$. The notebook then computes the following quantities:

    $\mathcal{I}_0[m_{\text{LLP}}]$, $\mathcal{I}_1[m_{\text{LLP}}]$, $\mathcal{I}_2[m_{\text{LLP}}]$, $\mathcal{I}_3[m_{\text{LLP}}]$

The integrals $\mathcal{I}_i$ are:

$$\mathcal{I}_0 = N_{\text{prod}}, \tag{5.2}$$

which shows the total number of LLPs produced at the given facility;

$$\mathcal{I}_1 = \frac{N_{\text{prod}}}{z_{\text{max}} - z_{\text{min}}} \int d\theta dE dz f_{\text{LLP}} \epsilon_{\text{az}}, \tag{5.3}$$

which shows the fraction of LLPs traveling in the direction of the decay volume times the length of the intersection of the LLP trajectory and the decay volume (normalized by the total number of the produced LLPs times the longitudinal length of the decay volume $z_{\text{max}} - z_{\text{min}}$);

$$\mathcal{I}_2 = \frac{N_{\text{prod}}}{c\tau_{\text{LLP}}} \int d\theta dE dz f_{\text{LLP}} \epsilon_{\text{az}} c\tau_{\text{LLP}} \frac{dP_{\text{dec}}}{dz}, \tag{5.4}$$

which is the fraction of the LLPs decaying inside the decay volume;

$$\mathcal{I}_3 = \frac{N_{\text{prod}}}{c\tau_{\text{LLP}}} \int d\theta dE dz f_{\text{LLP}} \epsilon_{\text{az}} c\tau_{\text{LLP}} \frac{dP_{\text{dec}}}{dz} \epsilon_{\text{dec}}, \tag{5.5}$$

which is the fraction of LLPs decaying inside the decay volume with the decay products satisfying the decay acceptance criteria. $\mathcal{I}_1$ shows the actual limitations on the sensitivity caused by the geometry of the experiments. $\mathcal{I}_2$ is the number of decays assuming that $\epsilon_{\text{dec}} \equiv 1$. $\mathcal{I}_3$ is the total number of events at the lower bound. For $\mathcal{I}_{2,3}$, the exporting is for $c\tau_{\text{LLP}} = 10^9$ m. The bound may be easily rescaled for an arbitrary $c\tau_{\text{LLP}}(m_{\text{LLP}}, g_{\text{LLP-SM}})$, where $g_{\text{LLP-SM}}$ is the LLP-SM coupling, given that these quantities scale as $(c\tau_{\text{LLP}})^{-1}$. In addition, when comparing the acceptances for various experiments, the dependence on $c\tau_{\text{LLP}}$ does not matter.

Basically, assuming that $N_{\text{ev}} \propto g^4$, where g is the LLP's coupling to the SM particles, the sensitivity at the lower bound is

$$g_{\text{lower}}^2 = \sqrt{N_{\text{min}} c\tau_{\text{LLP}}} / \sqrt{\mathcal{I}_4} \Big|_{g\,=\,1}, \tag{5.6}$$

where N_{min} is the critical number of events determining the boundary of the sensitivity region. These lower bounds estimates are also performed in the notebook.

5.2 Slow and fast methods to compute the number of events

The simplest way is to obtain the number of events (1.1) using the built-in Mathematica functions `Interpolation` and `NIntegrate`; this is the function `NeventsInt[m, coupling, ProdChannel]`, where `ProdChannel` is the given LLP's production channel. It is defined in the subsection *Number of events - slow*. The resulting brute-force integrals are very slow ($\mathcal{O}(1$ second$)$ per LLP's parameter space point "mass-coupling"). Given that the number of events has to be tabulated over a dense grid of LLP masses and couplings, the total computation time would be significant. In addition, the integration domain in θ, E, z may be much wider than the domain the events typically belong to. This is the case for the experiments where the detector covers a much narrower domain of the angles than the decay volume does. The Monte-Carlo integral (which randomly samples points inside the whole decay volume region) may return results with large errors in this situation.

To reduce the error and improve the speed of the computation, the subsection *Number of events - fast* introduces a much faster and more stable way of calculating the number of events using quick grid mapping (the function `NeventsDiscret[m, ProdChannel, couplingslist]`, where `couplingslist` is the list of couplings, e.g., $\{10^{-7\cdot}, 10^{-6\cdot}\}$).

The brute-force integral representation of the number of events serves to cross-check for the fast method (section *Tests*). Typically, the agreement between the two methods is within 10%. Larger discrepancies may happen in a few cases:

1. Rarely, the θ, E, z grid density assumed by default for the computation of `NeventsDiscret`, `OutGrid`θ`final`, `OutGridEfinal`, `OutGridzfinal`, is insufficient. To exclude this reason, the user may just increase the grid density from 30 to a higher value in these strings:

   `Do[{OutGrid`θ`final[prod], `$\Delta\theta$`vals[prod]} = OutGrid`θ`Temp[InGrid`$\theta\epsilon$`, 30, prod, `θ`maxBrem],`
   `{prod, ProductionList}]`

   `{OutGridzfinal, `Δ`zvals} = OutGridszTemp[InGridz`ϵ`, 30, zminExp];`

   to see whether the value of `NeventsDiscret` converges to `Nevents`.

2. A much more common reason is when the Monte-Carlo integration in `Nevents` fails to produce accurate results due to the abovementioned reasons.

5.3 What users have to provide

The users have to provide:

1. The tabulated LLP distribution functions and acceptances generated by the previous modules. At the very beginning of the notebook, it scans the folders

   spectra/New physics particles spectra/⟨LLP⟩

   Acceptances

   on the available tabulated angle-energy distributions and acceptances.

2. The probabilities of producing the LLP through various mechanisms, `ProbLLP[mLLP, coupling, prod]`, where `mLLP` is the LLP mass in GeV, `coupling` is the coupling in appropriate units, and `prod` must exactly match the name of the corresponding angle-energy distribution. Typically, this probability is the branching ratio of the decay (for the decay process) or the scattering production probability per proton-proton collision (for the production by, e.g., deep inelastic scattering). For already implemented LLPs, the tabulated probabilities are located in the folder

   phenomenology/⟨LLP⟩/branching ratios/

   The probability of producing the LLP through the given process `prod`, `ProbLLP[mLLP, coupling, prod]`, is typically defined by

   `ProbMother[prod]*PmotherToLLP[mLLP, coupling, prod,facility]`

   where `ProbMother` is the probability of having a mother particle per proton collision (defined in the chapter *Angle-energy distributions and differential yields for the given experiment*). If one considers `prod` as a decay of some mother particles X or the mixing with it, it is simply the fraction of the produced X per collision, `ProbMother` $= \sigma_{p\,\text{collision}\to X}/\sigma_{p\,\text{collision}}$. For the given experiment, it may be extracted from the `.mx` file with tabulated acceptances using the table `CrossSectionData` (assigned in the chapter *Cross-sections, acceptances*). By convention, if the mother particle is a charged particle, `CrossSectionData` includes a factor of 2 due to the charge-conjugated production channel. If one considers the scattering of protons (the proton bremsstrahlung, DIS, etc.), it is one by definition.

   They are imported and interpolated in the section *LLP phenomenology*, selected for the particular facility and convoluted with the production fractions of the mother particles (for the production via decays of secondaries) in sub-section *Importing distributions*.

6   4. Plots.nb

This small notebook simply scans the folder containing the sensitivities for the chosen LLP, lists all the available curves that match the specified minimum number of events and model-specific parameters, imports the selected curves, and makes the sensitivity plot.

7   EventCalc.nb

`EventCalc` is an additional module that generates detailed LLP event information. It does the same as `SensCalc`, except for decaying unstable decay products and propagating them through the detector. Its purpose is to pass the decay events to the full simulation framework of various experiments, where they have to be processed using detailed simulations.

The simplified logic of the module is as follows. Through the notebook, the user is subsequently asked about the experiment, the LLP, the production modes, and the decay modes. Then, it computes the tabulated distribution averaged over the selected production channels. Then, if the user selects the mass and the set of couplings describing the LLP, the routine

   `NeventsBlock[mass, coupling1, coupling2, Nevents]`

produces the total number of events and two tables `llpinfodata`, `decayproductsdata`. The first table is about the properties of the decaying LLP. Each row has the columns

$$\{p_{\text{LLP},x}, p_{\text{LLP},y}, p_{\text{LLP},z}, E_{\text{LLP}}, m_{\text{LLP}}, \cos(\theta_{\text{LLP}}), \tfrac{p_{\text{LLP}}}{m_{\text{LLP}}}, P_{\text{decay}}, x_{\text{decay}}, y_{\text{decay}}, z_{\text{decay}}, \epsilon_{\text{az}}, N_{\text{prod,tot}}, \epsilon_{\text{geom}}\}$$

Here, the first 5 columns are the 4-momentum and the mass of the LLP, the 6th and 7th are the cosine of the polar angle and the momentum-over-mass of the LLP, the 8th is the decay probability P_{decay}, the 9th-11th are the coordinates of the decay vertex, $N_{\text{prod,tot}}$ is the total number of produced LLPs at the given facility, and ϵ_{geom} is the azimuthal acceptance ϵ_{az} for the given LLP – the probability that the given LLP will have the azimuthal angle such that its trajectory intersects the decay volume. ϵ_{geom} and $N_{\text{prod,tot}}$ are constant for the given simulation, whereas P_{decay} and ϵ_{az} vary from event to event. The units are GeV for energies, momenta and mass, and meters for the decay coordinates.

The total number of events (that is automatically computed by the code `NeventsLLP[llpinfodata]`) is

$$N_{\text{events,total}} = N_{\text{prod,tot}} \times \epsilon_{\text{geom}} \times \frac{1}{N_{\text{simulated}}} \sum_{i=1}^{N_{\text{simulated}}} \epsilon_{\text{az}}^i \cdot P_{\text{decay}}^i, \tag{7.1}$$

where $N_{\text{simulated}}$ is the number of simulated events – the last argument in the `NeventsBlock` block.

The decay products table has the columns

$$p_x, p_y, p_z, E, m, \texttt{pdg, electric charge, stability}$$

The last two quantities are internal for `SensCalc` and may be dropped. Here, `stability` is 1 if the given particle is stable or metastable (such as muons, charged pions, or kaons), and 0 otherwise. `electric` charge for quarks is assumed to be an integer for internal purposes.

The block `NeventsBlock` also exports the two mentioned tables.

Currently, only dark photons are properly implemented in `EventCalc`. The other LLP models available in `SensCalc` will be added shortly after.

Some example of usage is provided in the subsection *Test* at the end of the notebook.